

Faire un formulaire HTML, CSS, PHP (1 sur 3)

I. Introduction au formulaire HTML CSS PHP

Dans ce tutorial, nous allons aborder la création de formulaires (x)html afin que vous soyez capable de créer un formulaire pour les pages de votre site web non seulement, mais aussi que ce formulaire respecte complètement les standards (x)html et utilise un fichier de style CSS pour la mise en forme. D'ailleurs nous allons compliquer un peu la mise en forme CSS pour que vous puissiez, je l'espère, apprendre des choses intéressantes en CSS.

Nous verrons ensuite comment traiter les données envoyées par ce formulaire, à titre d'exemple pour différents cas de figure, car les données récupérées dans un formulaire d'une page web peuvent être traitées de nombreuses façon suivant ce que l'on souhaite faire.

Un formulaire va se décomposer en trois parties : les balises **form** de début et fin du formulaire, les éléments dans lesquels les visiteurs vont entrer les informations (champs, listes, etc.), et un bouton submit (ou image) qui envoie les informations au serveur. Il y a aussi le traitement des données une fois l'envoi du formulaire effectué, cela est externe au formulaire mais sera aussi évoqué dans ce tutorial.

II. Explications

1. Aperçu du formulaire final

Voyons tout d'abord ce que vous allez apprendre à faire à travers ce tutoriel :



Coordonnées

6ma .FR
AIDE INFORMATIQUE

Civilité : M. Mlle Mme

Nom :

Adresse :

Code postal :

Ville :

Pays : Suisse

Centres d'intérêts : Sport Cinéma
 Internet Voyages

Message

Comme vous pouvez le voir, nous allons créer un peu de tous les types de champs les plus utilisés dans un formulaire pour que les formulaires HTML n'aient plus de secrets pour vous.

2. les balises du formulaire

Nous allons tout d'abord créer le formulaire avec la balise appropriée qui est **<form>**.

Pour créer un formulaire, il faut obligatoirement lui spécifier une méthode d'envoi d'une part (à choisir entre "get" et "post") ainsi que l'url du script qui va recevoir les données du formulaire. Celui se traduit par le code html :

```
php
texte
1. <form method="post" action="traitement.php">
2. </form>
```

Sur cet exemple, les données seront envoyées au fichier "traitement.php", dont nous verrons l'utilisation plus loin. Tous les éléments du formulaire seront placés entre la balise de début <form...> et la balise de fin </form>.

3. Création des blocs coordonnées et message

Comme vous l'avez vu sur l'image montrant le résultat final, nous allons diviser notre formulaire en deux blocs principaux : "coordonnées" et "message", avec un titre pour chaque bloc. Pour cela, nous allons créer deux éléments <p> sur lesquels nous allons prévoir une classe "titre" pour la mise en forme CSS par la suite.

Et nos deux groupes de champs vont être créés avec des éléments <fieldset> sur lesquels nous allons prévoir un attribut "id" pour la mise en forme plus tard.

Notre code devient :

```
php
texte
1. <form method="post" action="traitement.php">
2.   <p class="titre">Coordonnées</p>
3.   <fieldset id="coordonnees">
4.   </fieldset>
5.   <p class="titre">Message</p>
6.   <fieldset id="message">
7.   </fieldset>
8. </form>
```

Coordonnées



Message

4. Créer des input radio (case à choisir)

Pour le choix de la civilité, nous allons créer des boutons de type radio, c'est-à-dire à choix unique. Lorsque vous ferez un choix, le choix précédemment sélectionné ne le sera plus. Ils se créent avec la balise **<input type="radio"**, un nom **name="civilité"** et une valeur **value="valeur"**. Vous pouvez même utiliser **checked="checked"** sur une des cases si vous souhaitez qu'elle soit sélectionnée par défaut. Puis fermez la balise avec **</>**.

La valeur est ce que votre script de traitement du formulaire récupère, veillez donc à éviter tout caractère spécial...

Nous avons donc ajouté trois de ces cases (du même nom pour qu'un seul choix soit possible), précédé d'un élément **<label>**, qui est le label du champ comme son nom l'indique. Nous le mettrons pour chaque champ qui aura besoin d'une description le précédent. Enfin, on englobe tout cela d'un **<p>** avec un **id** que nous allons utiliser dans le CSS encore une fois, donc vous comprendrez son utilité plus tard. Notre code devient :

```
php
texte
1. <form method="post" action="traitement.php">
2. <p class="titre">Coordonnées</p>
3. <fieldset id="coordonnees">
4. <p id="civilite"><label>Civilité : </label>
5. <input type="radio" name="civilite" value="M." />M.
6. <input type="radio" name="civilite" value="Mlle" />Mlle
7. <input type="radio" name="civilite" value="Mme" />Mme
8. </p>
9. </fieldset>
10. <p class="titre">Message</p>
11. <fieldset id="message">
12. </fieldset>
13. </form>
```

Coordonnées

Civilité : M. Mlle Mme



Message

5. Créer les champs de texte

Les champs de texte sont les champs à une ligne dans lesquels nous pouvons entrer du texte : nom, adresse, code postal, ville sur l'exemple.

Ces champs de texte ne sont pas très compliqués à mettre en place, il faut tout d'abord créer un **<input type="text">**, puis attribuer un nom avec **name="nom"** et enfin une taille (longueur) avec **size="30"**. Mais une valeur par défaut est optionnellement possible avec **value="valeur"** et un nombre de caractères maximum avec **maxlength="100"** par exemple.

Dans notre formulaire, nous allons faire précéder chacun des quatre champs de texte d'un label puis faire un retour à la ligne après chaque champ avec **
**, ce qui donnera ceci :

```
php
texte
1. <form method="post" action="traitement.php">
2. <p class="titre">Coordonnées</p>
3. <fieldset id="coordonnees">
4. <p id="civilite"><label>Civilité : </label><br />
5. <input type="radio" name="civilite" value="M." />M.
6. <input type="radio" name="civilite" value="Mlle" />Mlle
7. <input type="radio" name="civilite" value="Mme" />Mme
```

```

8.   </p><br />
9.   <label>Nom : </label>
10.  <input type="text" name="nom" size="30" /><br />
11.  <label>Adresse : </label>
12.  <input type="text" name="adresse" size="30" /><br />
13.  <label>Code postal : </label>
14.  <input type="text" name="codepostal" size="30" /><br />
15.  <label>Ville : </label>
16.  <input type="text" name="ville" size="30" /><br />
17. </fieldset>
18. <p class="titre">Message</p>
19. <fieldset id="message"><br />
20. </fieldset>
21. </form>

```

Coordonnées

The screenshot shows a web form with the following fields:

- Civilité : M. Mlle Mme
- Nom :
- Adresse :
- Code postal :
- Ville :

The logo for '6ma AIDE INFORMATIQUE' is visible in the top right corner of the form area.

Message

The screenshot shows a single text input field for entering a message.

6. Créer une liste de choix

Nous allons ensuite créer une liste prédéfinie qui contiendra des pays, il faudra que l'utilisateur en choisisse un dans la liste.

Pour créer la liste il faut insérer une balise **<select>** sur laquelle sur mettrons un nom **name="pays"** qui nous servira à récupérer sa valeur pour le traitement du formulaire. Ensuite pour chaque option de liste, il faut créer une balise **<option>** avec une valeur **value="valeur"** et la fermer avec **</option>**. Entre les deux (**<option>** et **</option>**), il faut mettre la valeur de la liste affichée pour l'utilisateur. Puis en dernier lieu, il faut fermer la balise **</select>**.

Avec la liste rajoutée, on aura désormais ceci :

```

php
texte
1. <form method="post" action="traitement.php">
2. <p class="titre">Coordonnées</p>
3. <fieldset id="coordonnees">
4.   <p id="civilite"><label>Civilité : </label>
5.     <input type="radio" name="civilite" value="M." />M.
6.     <input type="radio" name="civilite" value="Mlle" />Mlle
7.     <input type="radio" name="civilite" value="Mme" />Mme
8.   </p>
9.   <label>Nom : </label>
10.  <input type="text" name="nom" size="30" /><br />
11.  <label>Adresse : </label>

```

```

12. <input type="text" name="adresse" size="30" /><br />
13. <label>Code postal : </label>
14. <input type="text" name="codepostal" size="30" /><br />
15. <label>Ville : </label>
16. <input type="text" name="ville" size="30" /><br />
17. <label>Pays : </label>
18. <select name="pays">
19. <option value="france">France</option>
20. <option value="belgique">Belgique</option>
21. <option value="suisse">Suisse</option>
22. </select>
23. </fieldset>
24. <p class="titre">Message</p>
25. <fieldset id="message">
26. </fieldset>
27. </form>

```

Coordonnées

Message

7. Créer des checkbox (cases à cocher)

Maintenant nous allons ajouter des cases à cocher. La différence avec les input de type radio, c'est que le choix peut être multiple ici. Vous pouvez cocher plusieurs cases comme bon vous semble. Il suffit de créer un élément `<input type="checkbox">` et comme nous récupérerons les valeurs cochées dans une variable de type tableau, nous allons toutes les nommer `name="interets[]"` et leur donner une valeur avec `value="valeur"`.

Le code avec notre partie contenant les cases à cocher pour les centres d'intérêts sera donc réalisé avec ce code :

```

php
texte
1. <form method="post" action="traitement.php">
2. <p class="titre">Coordonnées</p>
3. <fieldset id="coordonnees">
4. <p id="civilite"><label>Civilité : </label>
5. <input type="radio" name="civilite" value="M." />M.
6. <input type="radio" name="civilite" value="Mlle" />Mlle
7. <input type="radio" name="civilite" value="Mme" />Mme
8. </p>

```

```

9.   <label>Nom : </label>
10.  <input type="text" name="nom" size="30" /><br />
11.  <label>Adresse : </label>
12.  <input type="text" name="adresse" size="30" /><br />
13.  <label>Code postal : </label>
14.  <input type="text" name="codepostal" size="30" /><br />
15.  <label>Ville : </label>
16.  <input type="text" name="ville" size="30" /><br />
17.  <label>Pays : </label>
18.  <select name="pays">
19.    <option value="france">France</option>
20.    <option value="belgique">Belgique</option>
21.    <option value="suisse">Suisse</option>
22.  </select>
23.  <p id="interets"><label>Centres intérêts : </label>
24.    <input type="checkbox" name="interets[]" value="sport" />Sport
25.    <input type="checkbox" name="interets[]" value="cinema" />Cinéma<br />
26.    <input type="checkbox" name="interets[]" value="internet" />Internet
27.    <input type="checkbox" name="interets[]" value="voyages" />Voyages
28.  </p>
29. </fieldset>
30. <p class="titre">Message</p>
31. <fieldset id="message">
32. </fieldset>
33. </form>

```

Coordonnées

Message

8. Créer une zone de texte

Une zone de texte est une zone où vous pouvez taper du texte librement et sur plusieurs lignes, au contraire du champ de texte qui est limité et sur une seule ligne.

Sa création est assez simple, il faut créer une balise `<textarea>`, lui ajouter un nom avec `name="comments"` et éventuellement un nombre de lignes `rows="5"` et de colonnes `cols="40"`. Enfin il faudra fermer avec `</textarea>`.

Vous avez également la possibilité de mettre un texte par défaut dedans, pour cela le placer entre `<textarea...>` et `</textarea>`.

```
1. <form method="post" action="traitement.php">
2. <p class="titre">Coordonnées</p>
3. <fieldset id="coordonnees">
4.   <p id="civilite"><label>Civilité : </label>
5.     <input type="radio" name="civilite" value="M." />M.
6.     <input type="radio" name="civilite" value="Mlle" />Mlle
7.     <input type="radio" name="civilite" value="Mme" />Mme
8.   </p>
9.   <label>Nom : </label>
10.    <input type="text" name="nom" size="30" /><br />
11.   <label>Adresse : </label>
12.    <input type="text" name="adresse" size="30" /><br />
13.   <label>Code postal : </label>
14.    <input type="text" name="codepostal" size="30" /><br />
15.   <label>Ville : </label>
16.    <input type="text" name="ville" size="30" /><br />
17.   <label>Pays : </label>
18.    <select name="pays">
19.      <option value="france">France</option>
20.      <option value="belgique">Belgique</option>
21.      <option value="suisse">Suisse</option>
22.    </select>
23. <p id="interets"><label>Centres intérêts : </label>
24.   <input type="checkbox" name="interets[]" value="sport" />Sport
25.   <input type="checkbox" name="interets[]" value="cinema" />Cinéma<br />
26.   <input type="checkbox" name="interets[]" value="internet" />Internet
27.   <input type="checkbox" name="interets[]" value="voyages" />Voyages
28. </p>
29. </fieldset>
30. <p class="titre">Message</p>
31. <fieldset id="message">
32. <textarea name="comments" rows="5" cols="40"></textarea>
33. </fieldset>
34. </form>
```

Coordonnées



Civilité : M. Mlle Mme

Nom :

Adresse :

Code postal :

Ville :

Pays : ▼

Centres d'intérêts : Sport Cinéma
 Internet Voyages

Message

9. Créer des boutons submit et reset

Il ne nous reste plus qu'à créer un bouton pour envoyer le formulaire, qui déclenchera l'action du formulaire (souvenez-vous le fichier traitement.php ...). Pour cela il suffit de créer un **<input type="submit">** et de lui donner une valeur qui sera affichée sur le bouton avec **value="Envoyer"** par exemple, puis de fermer la balise bien sûr.

De la même façon nous pouvons aussi créer un bouton **<input type="reset">** qui servira à effacer tout le contenu des champs du formulaire pour recommencer à zéro. Nous mettrons alors la valeur **value="Recommencer"**. Nous allons mettre ces deux boutons dans un **<p> </p>** avec un id défini pour appliquer un style CSS dans la partie 2 du tutorial.

Voici maintenant le code complet du formulaire :

```
php
texte
1. <form method="post" action="traitement.php">
2. <p class="titre">Coordonnées</p>
3. <fieldset id="coordonnees">
4.   <p id="civilite"><label>Civilité : </label>
5.     <input type="radio" name="civilite" value="M." />M.
6.     <input type="radio" name="civilite" value="Mlle" />Mlle
7.     <input type="radio" name="civilite" value="Mme" />Mme
8.   </p>
9.   <label>Nom : </label>
10.    <input type="text" name="nom" size="30" /><br />
11.   <label>Adresse : </label>
12.    <input type="text" name="adresse" size="30" /><br />
```




```

13. <label>Code postal : </label>
14. <input type="text" name="codepostal" size="30" /><br />
15. <label>Ville : </label>
16. <input type="text" name="ville" size="30" /><br />
17. <label>Pays : </label>
18. <select name="pays">
19. <option value="france">France</option>
20. <option value="belgique">Belgique</option>
21. <option value="suisse">Suisse</option>
22. </select>
23. <p id="interets"><label>Centres intérêts : </label>
24. <input type="checkbox" name="interets[]" value="sport" />Sport
25. <input type="checkbox" name="interets[]" value="cinema" />Cinéma<br />
26. <input type="checkbox" name="interets[]" value="internet" />Internet
27. <input type="checkbox" name="interets[]" value="voyages" />Voyages
28. </p>
29. </fieldset>
30. <p class="titre">Message</p>
31. <fieldset id="message">
32. <textarea name="comments" rows="5" cols="40"></textarea>
33. </fieldset>
34. <p id="buttons">
35. <input type="submit" value="Envoyer" />
36. <input type="reset" value="Recommencer" />
37. </p>
38. </form>

```

Coordonnées



Civilité : M. Mlle Mme

Nom :

Adresse :

Code postal :

Ville :

Pays : ▼

Centres d'intérêts : Sport Cinéma
 Internet Voyages

Message

III. Conclusion du tutorial

Merci de votre attention. Vous venez de voir comment créer un formulaire HTML avec les principaux champs utilisés (champ de texte, case à cocher, liste, etc.). Mais pour le moment nous n'avons pas vu à quoi servent les **id** et **class** mis sur certains éléments, ni comment récupérer les données dans un script pour en faire quelque chose. La partie suivante va donc parler de [la mise en forme CSS du formulaire](#), et la troisième partie portera sur le [traitement de données du formulaire en PHP](#).

J'espère que vous aurez appris des choses à l'issu des ces trois tutoriaux, et si vous avez des questions, je vous répondrai avec plaisir sur le [forum](#).

Faire un formulaire CSS, HTML, PHP (2 sur 3)


I. Introduction au formulaire CSS

Ce tutorial fait suite à [Faire un formulaire HTML, CSS, PHP \(1 sur 3\)](#), donc commencez par cette partie 1 si vous souhaitez le suivre en entier.

Après avoir créé un formulaire en HTML, nous allons maintenant appliquer des styles CSS sur celui-ci afin de le mettre en forme, histoire qu'il soit un peu plus sexy ;-) mais aussi et surtout que vous appreniez le CSS.

Nous allons partir de cela...

Coordonnées



Civilité : M. Mlle Mme

Nom :

Adresse :

Code postal :

Ville :

Pays : ▼

Centres d'intérêts : Sport Cinéma
 Internet Voyages

Message

Envoyer

Recommencer

... pour arriver à cela :

Coordonnées

Civilité : M. Mlle Mme

Nom :

Adresse :

Code postal :

Ville :

Pays : Suisse

Centres d'intérêts : Sport Cinéma
 Internet Voyages

Message

6ma ^{FR}
AIDE INFORMATIQUE

II. Explications

1. Création du fichier CSS

Vous commencerez par créer une page html de base (avec votre éditeur html ou manuellement), ce qui donner ceci :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Document sans nom</title>
</head>

<body>
</body>
</html>
```

Ensuite, vous allez créer un fichier que vous nommerez "formulaire.css" dans le même répertoire que le fichier html et vous l'attacherez à cette page html, ce qui donnera :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Document sans nom</title>
<link href="formulaire.css" rel="stylesheet" type="text/css" />
</head>

<body>
</body>
</html>
```



Entre les balises **<body>** et **</body>**, vous placerez le code de notre formulaire créé dans la page 1 du tutorial. Nous allons maintenant voir le contenu du fichier CSS.

2. Les titres des blocs : coordonnées et message

Commençons par les éléments `<p class="titre">`. Nous allons leur mettre ceci :

- background : couleur de fond du titre
- color : couleur du texte
- padding : un petit espace pour ne pas écrire juste au bord du cadre du titre
- font-size : pour définir la taille du texte
- border : une bordure de 2 pixels avec un effet sortant (ombre) avec une couleur définie
- position : nous plaçons le titre en position relative à son conteneur
- margin-bottom : nous mettons une marge de bas négative, ce qui fait que le titre viendra s'encastrent dans l'élément en-dessous
- width : la largeur du cadre du titre
- margin-left : marge à gauche du titre
- margin-top : marge au-dessus du titre

```
• CSS
• texte
1. p.titre {
2.   background:#DED983;
3.   color:#345071;
4.   padding:.2em .3em;
5.   font-size:1.2em;
6.   border:2px outset #DED983;
7.   position:relative;
8.   margin-bottom:-1em;
9.   width:10em;
10.  margin-left:1em;
11.  margin-top:1em;
12. }
```

Ensuite nous allons configurer un peu les jeux de champs en appliquant un style sur tous les "fieldset" :

- border : on spécifie de ne pas mettre de bordures (on la mettra sur chaque jeu de champ au cas par cas)
- margin-bottom : marge du dessous

- width : la largeur des cadres de champs
- padding-top : espace en haut, pour laisser la place aux cadres de titres.

```

• CSS
• texte
1. fieldset {
2.   border:none;
3.   margin-bottom:1em;
4.   width:24em;
5.   padding-top:1.5em;
6. }

```

Puis un petit style général à toutes les listes déroulantes :

- margin-left : on spécifie une marge à gauche (pour laisser la place aux "labels" : légende des champs)
- margin-bottom : à 0, pas de marge en dessous.

```

• CSS
• texte
1. select {
2.   margin-left:9em;
3.   margin-bottom:0;
4. }

```

3. Les champs et labels du bloc coordonnées

A présent voyons le bloc (jeu de champs) "coordonnées" :

- background : nous allons lui spécifier une couleur de fond
- border : nous allons lui mettre une bordure "outset", donc qui va sortir derrière pour faire comme un effet d'ombre.

```

• CSS
• texte
1. /* fieldset coordonnees */
2. fieldset#coordonnees {
3.   background:#A4D8EE;
4.   border:outset #A4D8EE;
5. }

```

Puis définissons un style pour tous les labels du bloc coordonnées. Remarquez comment il est possible d'atteindre un sous-élément en CSS en spécifiant tout d'abord l'élément principal (ici dont l'id est "coordonnees"), puis un élément qui se trouve dans ce dernier (les label). Cela évite de devoir mettre des id ou class à tous les labels pour rien, d'autant qu'ils auront tous le même style :

- position : position absolue (relative à aucun autre élément)
- font-size : on spécifie une taille de police, ici 90% de la taille par défaut
- padding-top : un espace au-dessus du texte du label
- left : spécifie la position horizontale du coin haut gauche de l'élément (20 pixels)

```

• CSS
• texte
1. #coordonnees label {
2.   position:absolute;
3.   font-size:90%;
4.   padding-top:.2em;
5.   left:20px;
6. }

```

De la même façon, nous allons atteindre et mettre en forme tous les éléments de type input du bloc coordonnées :

- margin-left : une marge à gauche pour permettre de laisser de la place aux labels (légende des champs)
- line-height : on allonge l'interligne
- margin-bottom : on met une marge en-dessous pour ne pas que les champs se touchent.

```
• CSS
• texte
1. #coordonnees input {
2.     margin-left:9em;
3.     line-height:1.4em;
4.     margin-bottom:.2em;
5. }
```

4. Le bloc message

Maintenant dans le fieldset "message", cela sera plus vite fait puisque nous n'avons qu'une zone de texte. Nous allons juste :

- background : définition d'une couleur de fond pour le bloc
- border : mise en place d'une bordure "outset" (effet ombre sortante) comme pour le bloc précédent.

```
• CSS
• texte
1. /* fieldset message */
2. fieldset#message {
3.     background:#9DF2CE;
4.     border:outset #9DF2CE;
5. }
```

5. Les cases radio pour la civilité

Tout comme pour les labels, nous allons passer la taille de la police de caractères à 90% pour le texte des boutons radio :

```
• CSS
• texte
1. #civillite {
2.     font-size:90%;
3. }
```

Et mettre une marge à gauche comme pour les autres éléments input du formulaire, pour laisser la place aux légendes :

```
• CSS
• texte
1. #civillite input {
2.     margin-left:9em;
3. }
```

Le problème en faisant ce que nous venons de faire ci-dessus, c'est que la marge à gauche va être appliquée aux trois boutons radio...

Nous allons donc utiliser une technique que nous n'avons pas vu encore. Elle consiste à faire un : "#civillite input + input ", de façon à ce que le style soit appliqué uniquement si le champ input radio est précédé d'un autre. Futé non ? :-)

Si c'est le cas et que le champ input n'est pas le premier de la ligne, nous appliquons une marge beaucoup moins grande (1em).

```

CSS
texte
1. #civilite input + input {
2.     margin-left:1em;
3. }

```

6. Les cases à cocher pour les centres d'intérêt

Pour les cases à cocher, comme d'habitude nous commencerons à mettre la police à 90%, puis nous utiliserons les mêmes techniques que précédemment pour mettre une marge conséquente au premier élément (case à cocher), et moins grande pour les suivants (#interets input +input).

Ensuite vient quelque chose de nouveau mais que vous comprendrez assez facilement si vous avez compris le coup du "input +input ", c'est le "br+input". Le raisonnement est le même : s'il y a un élément

suivi d'un élément input, ce dernier aura une marge à gauche de 9em. Et oui car si l'on va à la ligne, il faut remettre la même marge que le premier input pour les aligner et laisser passer le label.

```

CSS
texte
1. /* cases a cocher des centres d'interet */
2. #interets {
3.     font-size:90%;
4. }
5. #interets input {
6.     margin-left:9em;
7. }
8. #interets input +input {
9.     margin-left:1em;
10. }
11. #interets br+input {
12.     margin-left:9em;
13. }

```



7. La zone de texte du message

Pour la zone de texte du message, voici le style que nous allons appliquer (il n'y a qu'un élément textarea donc nous ne spécifions pas d'id ni class) :

- font : la police de caractère ; sa taille sera 0.8em et elle sera de type "Trebuchet MS" (ou Verdana ou sans-serif si la ou les polices précédentes n'existent pas)
- width : la largeur de la zone de texte
- padding : nous allons espacer le texte de la zone de 0.2em de façon à ne pas écrire trop sur le bord.

```

CSS
texte
1. /* zone de texte du message */
2. textarea {
3.     font:.8em "Trebuchet MS", Verdana, sans-serif;

```

```
4. width:29em;
5. padding:.2em;
6. }
```

8. Les boutons submit et reset

Pour finir, nous allons voir les boutons submit et reset. Vous l'aurez peut-être remarqué, ils n'ont pas de "class" dans le code html, alors comment allons nous les atteindre ? Et bien nous allons utiliser leur type en faisant `input[type="submit"]` pour le bouton submit et la même chose avec le bouton reset. Nous allons leur appliquer le même style, à savoir :

- background : une couleur de fond
- font : une police de 1.2em en taille et de type "Trebuchet MS" (ou Verdana ou sans-serif si la ou les polices précédentes n'existent pas)
- color : une couleur du texte

NB : notez que l'on peut appliquer le même style à plusieurs éléments en les séparant d'une virgule comme ci-dessous.

```
• CSS
• texte
1. /* les boutons submit et reset */
2. input[type="submit"], input[type="reset"] {
3.   background:#DED983;
4.   font:1.2em "Trebuchet MS", Verdana, sans-serif;
5.   color:#345071;
6. }
```

Enfin nous allons centrer les deux boutons.

Attention : ils vont logiquement se mettre au centre de la page web et non au centre du formulaire, pour remédier à cela vous pouvez faire un div [conteneur](#) qui contient tous les éléments de notre page web, en lui spécifiant la largeur du formulaire. Si les boutons sont dedans ils s'aligneront alors sur le formulaire et non la page.

```
• CSS
• texte
1. p#buttons {
2.   text-align:center;
3. }
```

III. Conclusion du tutorial

Voilà, après avoir vu comment créer un formulaire HTML avec les principaux champs utilisés (champ de texte, case à cocher, liste, etc.), nous avons vu comment on peut mettre ceux-ci en forme avec du CSS. Vous remarquerez sur le résultat par l'alignement des éléments que l'on pourrait presque croire à l'utilisation d'un tableau HTML, mais il n'en est rien, tout est fait en CSS et ça rend tout de même le code un peu plus propre et facile à gérer.

Dans le tutorial suivant, nous allons voir comment [traiter les données qui sont envoyées par le formulaire, en PHP](#).

J'espère que vous aurez appris des choses à l'issue de ces trois tutoriaux, et si vous avez des questions, je vous répondrai avec plaisir sur le [forum](#).